INDEX

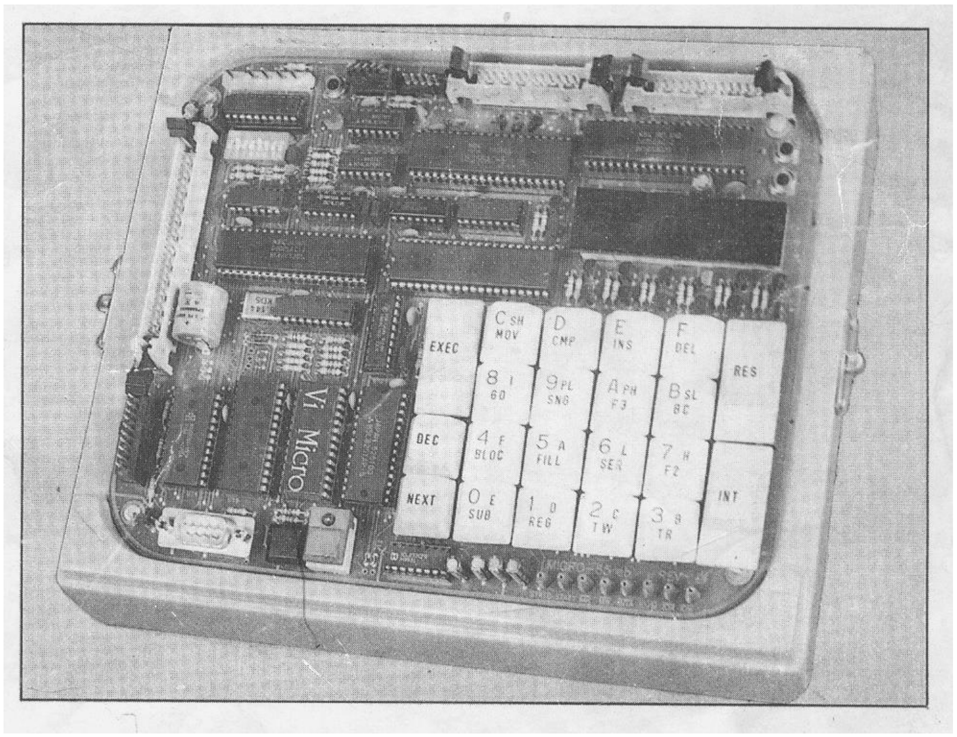
PROGRAM NO:1

AIM: STUDY OF INTEL 8085 MICROPROCESSOR KIT.

1.1 INTRODUCTION

This section briefs the hardware and software facilities available in both the trainers Micro-85 EBl and Micro-85 EB2. Micro-85 EBl is a powerful Microprocessor Trainer with basic features such as 24 TTL lines using 8255 Hardware Single Stepping and Software Single Stepping of user programs.

In addition to the above features, Micro-85 EB2 has RS232C compatible serial port, Bus Expansion for interfacing VBMB series of add-on cards and 24 TTL I/O lines. A separate switch is provided for learning more about hardware. interrupts. There is also provision to add multi output power supply for interfacing experiment boards. Most of the control



signals are terminated .at test points for easy analysis on CRO or logic probe.

The differences in the specification of Micro-85 EBl and Micro-85 EB2 are highlighted in this manual. The users are therefore requested to go through the Hardware specification carefully.

1.2 SPECIFICATIONS

1.2.1 HARDWARE SPECIFICATIONS

1) PROCESSOR & CLOCK FREQUENCY : Intel 8085A at 6.144 MHz clock. .

2) MEMORY:

Monitor EPROM	: 0000 - 1FFF
EPROM- Expansion	: 2000 - 3FFF & COOO - FFFF
System RAM	: 4000 - 5FFF
Monitor Data Area	: 4000 - 40FF (Reserved)
User RAM Area	: 4100 - 5FFF
RAM- Expansion	: 6000 - BFFF

Note: The RAM area from 4000 - 40FF should not be accessed by the user since it is used as scratch pad by the Monitor program.

3) INPUT/OUTPUT:

Parallel: 48 TTL :I/O lines using two number of 8255
(only 24 :I/O line. available in micro-85 EB1).

Serial : One number of RS232C compatible Serial interfaces
using 8251A - USART.

Timer : Three channel 16-bit Programmable Timer using 8253.
- Channel 0 is used as baud rate clock generator for 8251A USART.
- Channel 1 is used for in single stepping user programs.
- Channel 2 is used for Hardware Single Stepping user programs.

4) DISPLAY:

- 6 digit, 0.3", 7 Segment Red LED Display with filter.
- 4 digits for address display and 2 digits for data display.

5) KEYBOARD :

- 21 Keys soft keyboard including command keys and hexa-decimal keys.

6) AUDIO CASSETTE INTERFACE with file management.

7) BATTERY BACKUP:

- Onboard Battery backup facility is provided for the available RAM.

8) HARDWARE SINGLE STEP:

This facility allows the user to execute programs at machine cycle level using a separate switch.

9) SYSTEM POWER CONSUMPTION:

Micro-85 EB2	*Micro-85 EBl*
+ 5 V @ 1 Amp	+5V @ 500 mA
+ 12 V@ 200 mA	
- 12 V @ l00 mA	

10) POWER SUPPLY SPECIFICATIONS: *[External]*

Micro-85 EB2	**Micro-85 EB1**
Input: 230 V AC @ 50, Hz	230V AC @ 50 Hz
Output:+ 5 V @ 1.5 A	+ 5 V @ 600 mA
+ 12 V @ 150 mA	
- 12 V @ 150 mA	
+ 30 V @ 250 mA (Unregulated)	

11) PHYSICAL CHARACTERISTICS:

Micro-85 EB	PCB	: 230mm x 170mm [L x B]
	Weight	: 1 Kg.

12) TEST POINTS: Test points provided for MR*, MW*, INTA*, IO/M*,

IOR*,IOW*, S0, S1, INTA.

This enables the user to study the hardware timing easily.

1.2.2 SOFTWARE SPECIFICATIONS

Micro-S5 EB contains a high performance 8K bytes monitor program. It is designed to respond to user input, RS232C serial communications, tape interface etc. The following is a simple description of the key functions. Out of the 21 keys in the keyboard 16 are hexadecimal, command and register keys and the remaining are stand-alone keys.

KEY FUNCTION SUMMARY

$\boxed{\text{RES}}$ This *RES* key allows you to terminate any present activity and to return your Micro-85 EB to an initialized state. When pressed, the *µ..85* sign-on message appears in the display for a few seconds and the monitor will display command_prompt "-" in the left most digit.

$\boxed{\text{INT}}$ Maskable interrupt connected to CPU's **RST** 7.5 interrupt.

$\boxed{\text{DEC}}$ Decrement the address by one and display it contents

(or)

Display the previous register contents.

$\boxed{\text{EXEC}}$ Execute a particular program after selecting the address through GO command.

$\boxed{\text{DEC}}$ Increment address by one and display its contents

(or)

Display the next register content.

The 16 Hexa decimal keys have either a dual or a triple role to play.

i) It functions as a Hex key entry when a address or data entry is required.

ii) It functions as the Register key entry during Register command.

iii) It functions as command key when pressed directly after command prompt.

NOTE: The Hex-key function summary below is in the order:

i) Hex key. ii) Command key iii) Register key.

KEY *FUNCTION SUMMARY*

i. Hex key entry "0"

ii. This key is for substituting memory contents When NEXT key is pressed immediate1y after this it takes the user to the start address for

entering user programs, 4100 Hex (User RAM).

iii. Register key "E"

1
D

i. Hex key entry "1"

ii. Examine the 8085A registers and modify the same.

iii. Register key "D"

2
C

i. Hex key entry "2"

ii. Writes data from memory on to audiotape.

iii. Register key "C"

3
B

i. Hex key entry “3”

ii. Retrieve data from an audiotape to memory.

iii. Register key “B”

4
F

i. Hex key entry “4”.

ii. Block search for a byte.

iii. Register key “F”.

5
A

i. Hex key entry “5”.

ii. Fill a block of RAM memory with desired data.

iii. Register key “A”.

6
L

i. Hex key entry "6"

ii. Transmit/Receive data to/from the serial port.

The TW/TR keys are used for sending/receiving respectively.

iii. Register key “L”.

7
H

i. Hex key entry "7"

ii. Register key "H"

8
I

i. Hex key entry "8"

ii. Start running a particular program

iii. Register key "I"

9
PL

i. Hex key entry "9"

ii. Single step a program instruction by instruction.

iii. Register key “PCL”.

A
PH

i Hex key entry "A"

ii. Function key F3 F3 [0] = Input a byte from a port

P3 [1] = Output a byte to a port

iii. Register key “PCH"

iv. Used with SNG key for hardware single stepping.

B
SL

i Hex key entry "B"

ii. Check a particular block for blank.

iii. Register key "SPL"

C
SH

i. Hex key entry "C"

ii. Move block of memory to another block

iii. Register key “SPH”

D

i Hex key entry "D"

ii. Compare two memory blocks.

E

i. Hex key entry "E"

ii. Insert bytes into memory (RAM) .

F

i. Hex key entry "F"

ii. Delete bytes from memory (RAM) .

PROGRAM NO:2

AIM: - PROGRAM TO FIND ADDITION OF TWO 8 BIT NUMBERS

S NO.	MEMORY ADDRESS	MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
1	2000	21,01,24	LXI	H,2401	PLACE ADDRESS OF 1ST NUMBER IN H-L REGISTER PAIR.
2	2003	7E	MOV	A,M	MOVE THE CONTENTS OF MEMORY ADDRESSED BY H-L REGISTER PAIR TO THE ACCUMULATOR.
3	2004	23	INX	H	INCREMENT THE CONTENTS OF H-L REGISTER PAIR BY 1.
4	2005	86	ADD	M	ADD THE 2ND NO. IN THE 1ST NO. AND THE RESULT IN THE ACCUMULATOR.
5	2006	32,03,24	STA	2403	STORE RESULT IN MEMORY LOCATION 2403.
6	2009	76	HLT		STOP

OUTPUT

DATA

2401-48H

2402-56H

RESULT 2403-9EH

PROGRAM NO:3

AIM:- PROGRAM TO FIND SUBTRACTION OF TWO 8 BIT NUMBERS

S NO.	MEMORY ADDRESS	MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
1	2000	21,01,24	LXI	H,2401	PLACE ADDRESS OF 1ST NUMBER IN H-L REGISTER PAIR.
2	2003	7E	MOV	A,M	MOVE THE FIRST NUMBER IN ACCUMULATOR
3	2004	23	INX	H	ADD 1 TO THE CONTENTS OF H-L REGISTER PAIR.
4	2005	96	SUB	M	SUBTRACT 2ND NUMBER FROM THE FIRST NUMBER AND THE RESULT IS PLACED IN THE ACCUMULATOR.
5	2006	23	INX	H	GET MEMORY ADDRESS 2403 IN THE H-L PAIR
6	2007	77	MOV	M,A	MOVE THE CONTENTS OF ACCUMULATOR TO THE MEMORY LOCATION WHOSE ADDRESS IS IN THE H-L REGISTER PAIR.
7	2008	76	HLT		STOP

OUTPUT

DATA

48H

33H

RESULT

15H

PROGRAM NO:-4

AIM:- PROGRAM TO FIND 1'S COMPLEMENT OF A NUMBER.

S NO.	MEMORY ADDRESS	MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
1	2000		LXI	H,2401	PLACE ADDRESS OF 1ST NUMBER IN H-L REGISTER PAIR.
2	2003	7E	MOV	A,M	MOVE THE FIRST NUMBER IN ACCUMULATOR
3	2004	23	INX	H	ADD 1 TO THE CONTENTS OF H-L REGISTER PAIR.
4	2005	96	SUB	M	SUBTRACT 2ND NUMBER FROM THE FIRST NUMBER AND THE RESULT IS PLACED IN THE ACCUMULATOR.
5	2006	23	INX	H	GET MEMORY ADDRESS 2403 IN THE H-L PAIR
6	2007	77	MOV	M,A	MOVE THE CONTENTS OF ACCUMULATOR TO THE MEMORY LOCATION WHOSE ADDRESS IS IN THE H-

				L REGISTER PAIR.
7	**2008**	**76**	**HLT**	**STOP**

OUTPUT

DATA

48H

33H

RESULT

15H

PROGRAM NO:-5

AIM:-PROGRAM TO FIND 2'S COMPLEMENT OF A NO

S No.	MEMORY ADDRESS	MACHINE CODE	MNEMONICS	OPERANDS	COMMENTS
1	2000	3A,01,24	LDA	2401 H	LOAD THE NUMBER WHICH IS IN MEMORY 2401 H TO THE ACCUMULATOR
2	2003	2F	CMA		TAKE 1's COMPLEMENT OF THE CONTENTS OF ACC
3	2004	3C	INR	A	ADD TO THE CONTENTS OF ACC
4	2005	32,02,24	STA	2402H	STORE THE CONTENTS OF THE ACC IN THE M/M LOCATION 2402H
5	2008	76	HLT		STOP

OUTPUT

DATA

2401-78H

RESULT

2402-88H

PROGRAM NO:- 6

AIM:-PROGRAM TO FIND MULTIPLICATION OF TWO 8 BIT NUMBER.

S.NO.	M/M ADDRESS	MACHINE CODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
1	2000	2A,01,24		LHLD	2401H	PLACE MULTIPLICANT IN H LREGISTER PAIR .
2	2003	EB		XCHG		PLACE THE MULTIPLICAND IN THE REGISTER PAIR D-E.
3	2004	3A,O3,24		LDA	2403H	GET THE MULTIPLIER IN THE ACC.
4	2007	21,00,00		LXI	H,0000	PLACE THE INTIAL VALUE OF THE PRODUCT=00IN THE IN THE REGISTER PAIR H-L.
5	200A	0E,08		MVI	C,08	PLACE VALUE OF COUNT =08 IN IN REGISTER C .SINCE MULTIPLIER OF 8 BITS AND COUNT IS EQUAL TO THE BITS OF THE MULTIPLIER.
6	200C	29	LOOP	DAD	H	ADD THE CONTENTS OF H-L PAIR TO ITSELF.
7	200D	17		RAL		ROTATE CONTENT OF ACC WHICH IS MULTIPLIER,LEFT BY 1 BIT.
8	200E	D2,12,20		JNC	AHEAD	JUMP ON NO CARRY. IF MULTIPLIER BIT =1,NO,GO TO LABEL AHEAD
9	2011	19		DAD	D	IF MULTIPLIER BIT =1 THEN PRODUCT=PRODUCT+MULTIPLICANT
10	2012	0D	AHEAD	DCR	C	DECREMENT COUNT
11	2013	C2,0C,24		JNZ	LOOP	JUMP TO LABEL LOOP TILL C=0
12	2016	22,04,24		SHLD	2404H	STORE THE RESULT IN M/M LOCATION 404H AND 2405H
13	2019	76		HLT		STOP

OUTPUT

DATA

2401H-6,8

RESULT

2404H-48

PROGRAM NO:- 7

AIM:-PROGRAM TO FIND LARGEST NUMBER IN A DATA ARRAY.

S.NO .	**M/M ADDRE SS**	**MACHIN E CODE**	**LABEL**	**MNEM ONICS**	**OPERANDS**	**COMMENTS**
1	2000	21,00,24		LXI	H,2400	PLACE THE ADDRESS OF THE COUNT IN REGISTER PAIR H-L.
2	2003	4E		MOV	C,M	PLACE COUNT IN REGISTER C.
3	2004	23		INX	H	ADDRESS OF THE IST NO IS PLACED IN THE REGISTER PAIR H-L.
4	2005	7E		MOV	A,M	MOVE THE IST NO IN ACC
5	2006	0D		DCR	C	DECREMENT THE COUNT BY 1.
6	2007	23	LOOP	INX	H	PLACE ADDRESS OF THE NEXT NO. IN THE REGISTER PAIR H-L.
7	2008	BE		CMP	M	COMPARE THE NO. WHOSE ADDRESS IS IN H-L REGISTER PAIR WITH THE CONTENT OF ACC.
8	2009	D2,0D,20		JNC	AHEAD	JUMP NO CARRY, GO TO AHEAD, LARGER NO IN THE ACC.
9	200C	7E		MOV	A,M	IF CARRY GENERATED THEN PLACE THE NO. WHICH IN H-L PAIR TO THE ACC.
10	200 D	0D	AHEAD	DCR	C	DECREMENT COUNT BY 1
11	201 E	C2,07,20		JNZ	LOOP	JUMP NOT ZERO'IF THE COUNT IS IN REGISTER C IS NOT 0,GO TO INST'N LABELLED LOOP .
12	2011	32,50,24		STA	2450H	STORE THE LARGEST NO. IN LOCATION 2405H
13	2014			HLT		HALT

OUTPUT

DATA

2401-91H

2402-72H

2403-93H

RESULT

2405-93H

PROGRAM NO:- 8

AIM:-PROGRAM TO FIND SMALLER NUMBER IN A DATA ARRAY.

S. NO.	M/M ADDRESS	MACHINE CODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
1	2000	21,01,24		LXI	H,2401 H	PLACE THE ADDRESS OF THE NO. IN THE REGISTER PAIR H-L.
2	2003	7E		MOV	A,M	MOVE THE CONTENTS OF M/M ADDRESSED BY H-L PAIR.
3	2004	23		INX	H	ADD 1 TO THE CONTENTS OF H-L PAIR,H-L PAIR NOW CONTAINS 2402.
4	2005	BE		CMP	M	COMPARE THE 2ND NO. WITH THE IST NO.
5	2006	DA,OA,20		JC	AHEAD	JUMP ON CARRY.IF CARRY IS GENERATED,JUMP TO LABEL AHEAD.
6	2009	7E	AHEAD	MOV	A,M	IF NO CARRY GENERATED THEN MOVE THE CONTENTS OF M IN THE ACC.
7	200A	32,50,24		STA	2450 H	STROE THE SMALLER NO. IN LOCATION 2450H.
8	200 D	76		HLT		STOP

OUTPUT

DATA

2401-56H

2402-69H

RESULT

2405-56H

PROGRAM NO:- 9

AIM:-PROGRAM TO FIND SQ. ROOT OF 8 BIT NO.

S NO.	M/M ADDRESS	MACHINE CODE	LABEL	MNEM ONICS	OPERANDS	COMMENTS
1	2000	0E,01		MVI	C,01	PLACE 01 IN REGISTER C.
2	2002	06,01		MVI	B,01	PLACE ODD NO. 1 IN THE REGISTER B.
3	2004	3E,24		MVI	A,24	LOAD THE ACC WITH THE GIVEN NO.
4	2006	90	LOOP	SUB	B	SUBTRACT THE ODD NO. FROM THE ACC.
5	2007	CA,10,20		JZ	AHEAD	IF ACC CONTENTS ARE 0,THEN GO TO LABEL AHEAD.
6	200A	0C		INR	C	INCREMENT REGISTER C.
7	200B	04		INR	B	INCREMENT ODD NO.
8	200 C	04		INR	B	INCREMENT ODD NO.
9	200 D	C3,06,20		JMP	LOOP	REPEAT SUBTRACTION.
10	2010	79	AHEA D	MOV	A,C	MOVE THE CONTENTS OF C TO ACC.
11	2011	32,50,20		STA	2050	STORE THE RESULT IN M/M LOCATION 2050.
12	2014	76		HLT		STOP

OUTPUT

DATA

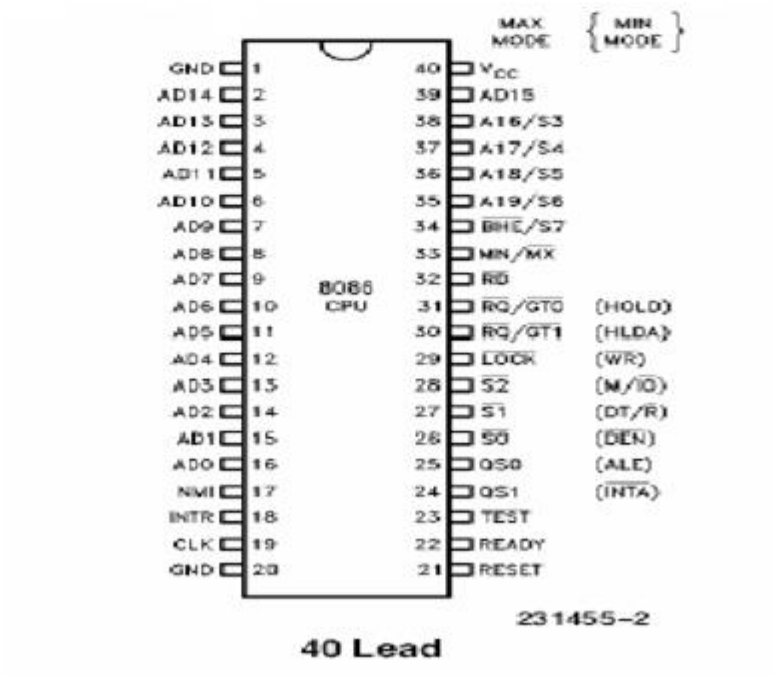
SQ. ROOT OF 36

RESULT

6

PROGRAM NO:-10

AIM: - STUDY OF INTEL 8086 MICROPROCESSOR



1) 16-bit Data bus2.
2) Computes 16 bit / 32 bit data.3.
3) 20-bit address bus.4.
4) More memory addressing capability (220= 1MB)
5) 16 bit Flag register with 9 Flags
6) Can be operated in Minimum mode and Maximum mode.
7) Has two stage pipelined architecture.
8) No internal clock generation.

9) 40 pin DIP IC - HMOS technology.

10) Operates on +5V supply voltage.

11) Has more powerful instruction set

Memory:
Program, data and stack memories occupy the same memory space. The total addressable memory size is 1MB KB. As the most of the processor instructions use16-bit pointers the processor can effectively address only 64 KB of memory. To access memory outside of 64 KB the CPU uses special segment registers to specify where the code, stack and data 64 KB segments are positioned within 1 MB of memory (see the "Registers" section below).

16-bit pointers and data are stored as :address : low-order byteaddress+1 : high-order byte32-bit addresses are stored in "segment:offset" format as :address : low-order byte of segmentaddress+1 : high-order byte of segmentaddress+2 : low-order byte of offsetaddress+3 : high-order byte of offset Physical memory address pointed by SEGMENT:OFFSET pair is calculated as:
Physical address = (<Segment Addr> * 16) + <Offset Addr>Program memory
- program can be located anywhere in memory. Jump and call instructions can be used for short jumps within currently selected 64 KB code segment, as well as for far jumps anywhere within 1 MB of memory. All conditional jump instructions can be used to jump within approximately +127 - -127 bytes from current instruction.
Data memory
- the processor can access data in any one out of 4 available segments, which limits the size of accessible memory to 256 KB (if all four segments point to different 64 KB blocks). Accessing data from the Data, Code, Stack or Extra segments can be usually done by prefixing instructions with the DS:,CS:, SS: or ES: (some registers and instructions by default may use the ES or SS segments instead of DS segment).Word data can be located at odd or even byte boundaries. The processor uses two memory accesses to read 16-bit word located at odd byte boundaries. Reading word data from even byte boundaries requires only one memory access.
Stack memory
can be placed anywhere in memory. The stack can be located at odd memory addresses, but it is not recommended for performance reasons (see"Data Memory" above).
Reserved locations

0000h - 03FFh are reserved for interrupt vectors. Each interrupt vector is a32-bit pointer in format SEGMENT:OFFSET.

FFFF0h - FFFFFh - after RESET the processor always starts program execution at the FFFF0h address.

Interrupts:

The processor has the following interrupts:

INTR

is a maskable hardware interrupt. The interrupt can be enabled/disabled using STI/CLI instructions or using more complicated method of updating the FLAGS register with the help of the POPF instruction. When an interrupt occurs, the processor stores FLAGS register into stack, disables further interrupts, fetches from the bus one byte representing interrupt type, and jumps to interrupt processing routine address of which is stored in location 4 * <interrupt type>. Interrupt processing routine should return with the IRET instruction.

NMI

is a non-maskable interrupt. Interrupt is processed in the same way as the INTR interrupt. Interrupt type of the NMI is 2, i.e. the address of the NMI processing routine is stored in location 0008h. This interrupt has higher priority then the maskable interrupt.

Software interrupts

can be caused by:

INT instruction - breakpoint interrupt. This is a type 3 interrupt.

INT <interrupt number> instruction - any one interrupt from available 256interrupts.

INTO instruction - interrupt on overflow

Single-step interrupt - generated if the TF flag is set. This is a type 1interrupt. When the CPU processes this interrupt it clears TF flag before callingthe interrupt processing routine.

Processor exceptions: divide error (type 0), unused opcode (type 6) andescape opcode (type 7).Software interrupt processing is the same as for the hardware interrupts.

I/O ports:

8086 can interface maximum of 65536 nos of 8-bit I/O ports. These ports can bealso addressed as 32768 16-bit I/O ports.

Registers:

Most of the registers contain data/instruction offsets within 64 KB memory segment. There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses four segment registers:

Code segment

(CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions.

Stack segment

(SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment.SS register can be changed directly using POP instruction.

Data segment

(DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

Extra segment

(ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions. It is possible to change default segments used by general and index registers by prefixing instructions with a CS, SS, DS or ES prefix.

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are:

Accumulator (AX)

register consists of 2 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations and string manipulation.

Base (BX)

register consists of 2 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

Count (CX)

register consists of 2 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low-order byte of the word, and CH contains the high-order byte. Count register can be used as a counter in string manipulation and shift/rotate instructions.

Data (DX)

register consists of 2 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low-order byte of the word, and DH contains the high-order byte. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

The following registers are both general and index registers:

Stack Pointer

(SP) is a 16-bit register pointing to program stack.

Base Pointer

(BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index

(SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data address in string manipulation instructions.

Destination Index

(DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data address in string manipulation instructions. Other registers:

Instruction Pointer

(IP) is a 16-bit register.

Flag Register

It is a 16-bit register containing 9 no.s of one bit flags:

- Overflow Flag (OF) - set if the result is too large positive number, or is too small negative number to fit into destination operand.
- Direction Flag (DF) - if set then string manipulation instructions will auto-decrement index registers. If cleared then the index registers will be auto-incremented
- Interrupt-enable Flag (IF) - setting this bit enables maskable interrupts.
- Single-step Flag (TF) - if set then single-step interrupt will occur after the next instruction.
- Sign Flag (SF) - set if the most significant bit of the result is set.
- Zero Flag (ZF) - set if the result is zero.
- Auxiliary carry Flag (AF) - set if there was a carry from or borrow to bits 0-3in the AL register.
- Parity Flag (PF) - set if parity (the number of "1" bits) in the low-order byte of the result is even.
- Carry Flag (CF) - set if there was a carry from or borrow to the most significant bit during last result calculation.

Instruction Set:

8086 instruction set consists of the following instructions:

- Data moving instructions.
- Arithmetic - add, subtract, increment, decrement, convert byte/word and compare.
- Logic - AND, OR, exclusive OR, shift/rotate and test.
- String manipulation - load, store, move, compare and scan for byte/word.
- Control transfer - conditional, unconditional, call subroutine and return from subroutine.
- Input/Output instructions.